

# Pattern Based Melody Matching Approach to Music Information Retrieval

<sup>1</sup>D.Vikram and <sup>2</sup>M.Shashi

<sup>1,2</sup>Department of CSSE, College of Engineering, Andhra University, India

<sup>1</sup>daravikram@yahoo.co.in, <sup>2</sup>smogalla2000@yahoo.com

## ABSTRACT

Digitization of music and advancements in information technology for sharing information on World Wide Web paved way for its availability in enormous quantities anywhere any time. Rather than retrieving annotated music in response to query given in terms of Metadata such as name of the composer/singer, genre etc modern researchers are challenged towards content based music information retrieval systems (CBMIR). CBMIR systems differ while representing the main melody either as a note sequence or as an analog acoustic signal; the note sequence representation is explored in this research work. Based on the observation that repeating patterns of the note sequences representing the main melody capture the essence of the music object, this research work developed a framework to investigate the feasibility and effectiveness of pattern based melody matching approach to music information retrieval. Experimentation is conducted on a real world dataset of musical objects belonging to South Indian classical music and the performance of the framework is estimated in terms of Mean Reciprocal Ranking.

**Keywords** – CBMIR, DTW, MIDI, MRR, QBH.

## 1 Introduction

Popular music objects are digitized and hence are available anywhere and anytime. Due to the huge volume of available music objects, retrieval of a specific object based on the user request is becoming more and more complex. Existing Music Information Retrieval Systems [2] accept user request expressed as a logical combination of metadata items like title, composer name, genre, singers name, movie name in terms of which the music databases are indexed and maintained. Such representations of musical objects cannot support content based retrieval of music objects as they are limited to matching based on metadata. However the researchers are challenged by the demand to retrieve music in response to query given in terms of content rather than metadata leading to content based representation of music objects. In the context of information retrieval the content of a music object is often captured by its main melody.

The main melody of a music object is a series of musical notes (semi tones in an octave) played whose frequencies are represented in terms of MIDI note numbers. The musical frequencies are divided into 11 octaves numbered from '-1' to '9' each containing 12 semitones named 'C C# D D# E F F# G G# A A# B'. The range of frequencies [11] encompassed by an octave doubles as you go for higher octaves successively. The name of the note reflects the semitone and the octave like 'A4' represents semitone 'A' in 4th octave and it has a distinct MIDI note number '60'. The MIDI note number is determined by the

following formula that transforms hummed tune frequency into the representation of MIDI values (semitones):

$$\text{MIDI Value} = 69 + [12 \times \log_2 \left( \frac{\text{freq}}{440} \right)]$$

Where *freq* is the frequency of hummed note and the operator '[' ]' calculates the nearest integer value, 12 leads to the classic dodecaponic musical scale, and 69 is the MIDI note number that corresponds to central A with pitch equal to 440 Hz. By convention middle C (MIDI note Number 60) is C4. A MIDI note number of 69 is used for A 440 tuning, that is the note A above middle C.

Content Based Music Information Retrieval (CBMIR) Systems rely on the frequent patterns extracted from the main melody for identifying the music objects in response to query given in terms of content referred to as "Content Query". Such a retrieval scenario belongs to query-by-example paradigm in which deals two types of content queries namely Query by Patterns and Query by Humming. In the context of Query by Patterns the CBMIR finds the music objects that contain repeated occurrences of the query patterns and ranks them based on the prominence of query patterns in the music objects. The second type of content query called QBH [3,6] refers to accepting user's music requirement in the form of audio input representing a part of the required song either sung or hummed at the user interface.

The main melody of the query should be represented as a note sequence and segmented into phrases before proceeding for matching. Irrespective of the way the query is received, a CBMIR have to deal with approximate matching of patterns/phrases [4] while building the database as well as responding to a query. Hence modern MIRs have to adopt intelligent pattern extraction and matching techniques in support of content based music information retrieval [7,8,14]. In this paper the authors proposed an effective framework for CBMIR is synthesized applying various concepts and techniques of sequential mining, approximate pattern mining, DTW and other relevant concepts of Content Based Information Retrieval.

The Mean Reciprocal Ranking (MRR) [1] is found suitable to assess the performance of the system for different inputs. Section 2 presents the recent research outcomes on various issues of CBMIR. Section 3 discusses the experimentation and results followed by conclusion.

## 2 Related Work

Anssi Klapuri [5] discusses various methods for dividing musical audio signal into shorter sequences if they occur repeatedly applying pattern segmentation and clustering approaches. A Self Distance Matrix (SDM) is often used for audio based analysis of segments of music expressed as acoustic signals. Repeated sequences occur as off-diagonal-strips of SDM and hence they are recognized.

Rifki Afinai Putri [1] et al discussed the representation of main melody as note sequence to transform 250 Indonesian pop songs in MIDI format and used DTW for matching the query with the whole song to take care of imperfect queries. The Query by Humming System could retrieve appropriate songs provided there is no difference in sruthi (key) of query and the song.

Alexios [9] et al proposed a subsequence matching algorithm for identifying a matching subsequence for a given query from a large database of music objects. The paper highlights the effectiveness of the method for a query by humming application.

### 3 Methodology

In this research a framework is developed for Content Based Music Information Retrieval Systems which consists of five modules namely main melody representation as note sequences, finding approximate repeating patterns, and pattern based indexing of music objects, segmentation of the input query and pattern matching and ranking based on the relevance of music objects. Figure.1 depicts the components of the proposed framework.

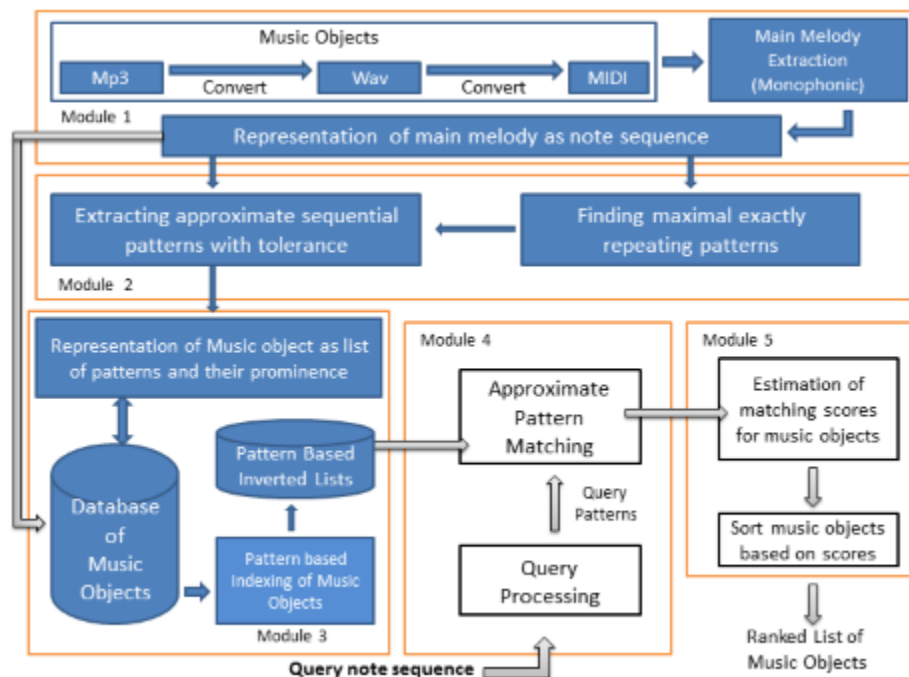


Figure.1: Framework of Proposed Content Based Music Information Retrieval System

Representation of a music object as a note sequence, after converting it into MIDI file format and extracting the main melody was discussed in detailed in [10]. Based on the user specified or predetermined thresholds on number of repeats of a pattern and tolerance of approximate matching, the module 2 of the framework extracts approximate sequential patterns by assembling the exact repeating patterns possibly separated by a tolerable gap. The details of pattern extraction from the music objects are discussed in [11]. Module 3 represents the music objects in terms of constituent repeating patterns along with their prominences to construct a pattern base of inverted lists which are used for estimating the matching scores of music objects. Module 4 preprocesses query to extract query patterns and finds matching patterns from the pattern base. The matching scores of music objects to the query are estimated to produce a ranked list of music objects in module 5. This paper discusses the implementation of the last three modules of the framework.

This framework adopts proven techniques of document retrieval for content based music information retrieval by appropriately tuning the required metrics and methods. Analogous to the words in the document text, repeating patterns of the note sequence representing the main melody capture the essence of the music object and hence the proposed framework explores pattern based melody matching approach to music information retrieval.

The note sequence representing the main melody of a music object needs to be transformed into a list of constituent repeating patterns which calls for complex asynchronous periodic pattern mining. Similar to the concept of document representation as a term vector, each music object is represented in terms of constituent repeating patterns along with their prominence estimated using a variant of *tf.idf*. It is essential to allow a certain level of tolerance while matching musical patterns as against exact matching of words/terms used in the context of document retrieval. Hence approximate pattern matching techniques are implemented using Dynamic Time Warping (DTW) algorithm for matching query patterns with the repeating patterns of the vocabulary developed for the corpus. In the context of query by example the cardinality of the result set for a query is much smaller, if not one, compared to the document retrieval scenario. Hence the performance metrics for CBMIR systems differ from those of document retrieval systems.

The first three modules of the framework prepare the database of music objects for intelligent content based retrieval which is performed off-line and is represented in dark colored blocks in Figure.1. Module 3 receives the set of repeating sequential patterns along with their frequency of occurrence in a song as the outcome of module 2 which applied the approximate sequential pattern mining algorithm on the note sequence representing the song/music object. Module 3 collects the repeating approximate patterns occurred in all songs/ musical objects of the corpus and maintains them as vocabulary of the corpus. It counts "*sf(p)*", the number of songs covering each pattern, *p*, of the vocabulary to estimate inverse song frequency similar to the concept of inverse document frequency. The prominence of a pattern, *p*, in a music object *s* is estimated as a product of its proportionate frequency in the song and the inverse song frequency as given below:

$$\text{Prominence (p, s)} = \text{frequency of p in } \frac{s}{|s|} * \text{Loge}\left(\frac{N}{sf(p)}\right)$$

Each musical object is represented as a vector of patterns and their prominences analogous to the concept of document representation as a term vector.

For fast retrieval in response to a query, a pattern base is created as an array of vectors to maintain inverted list of songs covering each pattern. Each element of the array contains the list of ordered pairs of the form *<song\_id, prominence>* representing the list of songs containing the pattern along with the prominence of the pattern in the song. This Pattern Base contains 1 to *n* vectors and the *i*<sup>th</sup> vector of the array is the list of songs containing *i*<sup>th</sup> pattern. The last three columns of Table 2 depict a portion of the pattern base.

### 3.1 Query Processing

The query consists of fragments of a song or humming that corresponds to a part of a song which is available in the database of music objects. Each query is converted into a note sequence before extracting repeating patterns from it referred to as query patterns in module 4.

Each query pattern will be compared with the patterns in the pattern base to identify matching patterns, if exists; the matching scores of the songs in the corresponding inverted list of each matching pattern will be incremented by the prominence of the pattern given in the inverted list. This process continues for each query pattern incrementing the matching scores of appropriate songs as guided by the inverted list.

In the context of CBMIR, the query given by the user as a piece of main melody may not be perfect due to altered "gamakas". Hence the query patterns are expected to contain additions, deletions or substitutions of musical notes from the corresponding patterns existing in the pattern base. Dynamic Time Warping (DTW) algorithm is found suitable [1,12] for identifying matching patterns for a given query pattern as it considers possible insertion, deletion and replacement of symbols contained in symbolic sequences of different lengths. DTW algorithm finds the distance between two symbolic sequences by the best possible alignment through maximal matching. Let X and Y be symbolic sequences of length n and m respectively.  $D(i,j)$  represents the distance between subsequences representing i and j long prefixes of X and Y respectively.  $d(i,j)$  is the distance between the pair of symbols i and j. DTW algorithm is described below.

Input: Given two sequences  $X=(x_1,x_2 \dots x_i \dots x_n)$  and  $Y=(y_1, y_2 \dots y_j \dots y_m)$

Output: Distance between X and Y,  $D(n,m)$

### 3.2 Algorithm 1 for Estimating the distance between query sequence and data sequence

1. Estimate the distance between every pair of distinct symbols constituting X and Y
2. Create distance matrix with rows corresponding to symbols of X and columns corresponding to symbols of Y so that  $d(i, j)$  represents the local distance between  $x_i$  and  $y_j$

for  $i \in [1, n]$  and  $j \in [1, m]$

3. Initialize the matrix D:

$$D(1, 1) = d(1, 1)$$

$$\text{for } i \in [2, n], D(i, 1) = D(i - 1, 1) + d(i, 1)$$

$$\text{for } j \in [2, m], D(1, j) = D(1, j - 1) + d(1, j)$$

4. Compute the remaining elements of matrix D

for i from 2 to n

for j from 2 to m

$$D(i,j)=\text{Min}\{D(i-1,j), D(i-1,j-1), D(i,j-1)\}+d(i,j)$$

5. Return  $D(n, m)$ .

#### Example:

Query Sequence: 3 3 3 5 1 1 1

Consider a note sequence.

Sequence : 3 3 5 5 5 1 1

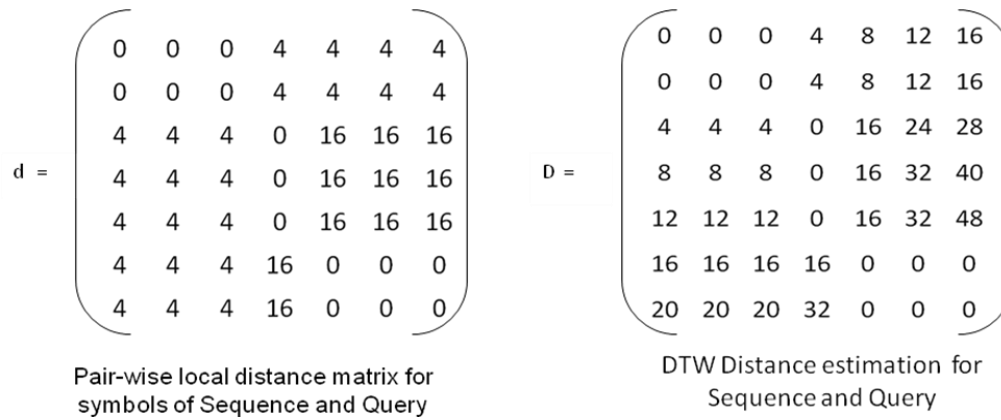


Figure.2: Matching Query sequence and the Sequence in Pattern Base

DTW algorithm is used to find the distance between query pattern and the patterns in the pattern base. The patterns whose distances from the query pattern are less than a given threshold are selected as matching patterns. The query pattern is skipped if the distance to the closest pattern is more than the threshold.

### 3.3 Upon Receiving a Query:

Module 4 receives the query which will be processed in a similar manner like any music object described in module 1 to extract query patterns. Each query pattern is matched with the patterns in the pattern base allowing certain level of tolerance (based on a threshold) using DTW algorithm.

### 3.4 Algorithm 2 for query processing online:

Input: Query as a note sequence with tolerance threshold for matching

*Step 1:* Query pattern extraction: Extract patterns in the query either by segmentation or by approximate pattern extraction algorithm depending on the context.

*Step 2:* Create an array of matching scores whose size is equal to the number of songs in the corpus and initialize its elements to zero.

# The  $i^{\text{th}}$  element in this array maintains the matching score of the  $i^{\text{th}}$  song to the query.

Repeat Steps 3 and 4 for each query pattern.

*Step 3:* Apply DTW on successive elements of the pattern base to find approximately matching patterns for a given query pattern.

*Step 4:* Repeat for each matching pattern,  $j$

Process the ordered pairs contained in the ' $j^{\text{th}}$ ' vector of the Pattern Base one by one

as detailed below:

Add ' $p$ ', the prominence of ' $j^{\text{th}}$ ' pattern in song,  $s$ , as mentioned in the ordered pairs  $\langle s, p \rangle$  to the ' $s^{\text{th}}$ ' entry of the array of matching scores.

Outcome: Array of scores incremented to finally reflect the matching scores of various songs for a given query.

**Step 5:** Sort the songs based on the matching scores to generate ranking list of songs for the query.

The results of melody matching and ranking is illustrated with an example in terms of Table 1, 2 and 3. The pattern base contains musical note sequences repeatedly occurred in various songs along with their prominence. In the example the given query contained three patterns as shown in Table 1. Each query pattern has one or more matching patterns found in the pattern base by applying DTW algorithm. Table 2 depicts for each query pattern the list of matching patterns along with the song(s) in which it occurred with the prominence given in the last column. As described in the algorithm step 4 the matching scores of pertinent songs are calculated and shown in Table3 as a ranking list of songs in response to the query.

1. List of Query Patterns Identified

**Table.1**

S. No.	Pattern
1	G3G3G3G3D3D3D3D3C3C3C3C3
2	D#3D#3C3C3D3D3C3C3C3C3
3	C3C3D3D3C3C3C3C3D3D3

2. For the given query patterns details of matching patterns, songs covering patterns, and prominence:

**Table. 2**

S. No	Query pattern	Matching Pattern	Song Name	Prominence
1	G3G3G3G3D3D3D3D3C3C3C3C3	G3G3G3G3D3D3D3D3C3C3C3C3	Aadikondar	0.132373291
2	D#3D#3C3C3D3D3C3C3C3C3	D#3D#3C3C3D3D3C3C3C3C3	Aadikondar	0.132373291
		D#3D#3C3C3D3D3C3C3C3C3C3C3D3D3D3D3C3C3	Renukadevi	0.218312283
3	C3C3D3D3C3C3C3C3D3D3	D3D3C3C3D3D3C3C3C3C3D3D3	Jaya Jaya Swamin	0.112029863
		C3C3D3D3C3C3C3C3D3D3C3C3	Rajarajeswari Choornika II	0.116794617
		C3C3D3D3C3C3D3D3C3C3C3C3D3D3	Swaminatha Paripalayasumam	0.112321891
		D3D3D3D3C3C3D3D3C3C3C3C3D3D3	Aadikondar	0.132373291
		D3D3C3C3D3D3C3C3D3D3C3C3C3C3D3D3	Janani Janani	0.213577162
			Thirumandiram 1	0.209926684
		D3D3C3C3D3D3C3C3C3C3D3D3	Dhina Vazhipadu aadalgal	0.096345908
		C3C3D3D3C3C3C3C3D3D3#3D#3G3G3	Aadikondar	0.132373291
		C3C3D3D3C3C3C3C3D3D3	Mayiladum Mayilayin Karpagme	0.204650562

3. Ranked List of Songs:

**Table. 3**

Songs	Matching Score
Aadikondar	0.52949
Renukadevi	0.21831
Janani Janani	0.21358
Thirumandiram 1	0.20993
Mayiladum Mayilayin Karpagme	0.20465
Rajarajeswari Choornika II	0.11679
Swaminatha Paripalayasumam	0.11232
Jaya Jaya Swamin	0.11203
Dhina Vazhipadu Paadalgal	0.09635



## 4 Experiments and Results

### 4.1 Dataset

Raga Surabhi [13] provides a collection of 700 (approximately) songs/musical objects belonging to various ragas of Carnatic music (South Indian Classical Music) including ragaalapana, songs and signature of each raga in mp3 format. Each song is represented as a note sequence during the preprocessing steps by converting wave files into strings. The length of the songs varies extensively resulting in a range of 238 to 6144 long note sequences. 17250 repeating patterns were extracted from these songs and a pattern base is maintained with inverted lists.

50 songs were selected for test set and query patterns from these test songs were given as input for retrieving ranked list of songs containing matching patterns to the query patterns. Mean Reciprocal Ranking (MRR) is a metric suitable for finding the accuracy of query-by-example based CBMIR systems aiming at a single song to be retrieved in response to a given query.

Let  $S(q_i)$  be the song aimed at and hence containing the query pattern(s) constituting the query  $q_i$  and let  $|Q|$  be number of queries used in the testing process, then

$$\mathbf{MRR} = \frac{1}{|Q|} \sum_i \frac{1}{\text{Rank}(S(q_i))}$$

Experiments were conducted on the test set to estimate MRR for each query length by extending the queries (aiming at each song of the test set) from 1 to 5 query patterns. It was observed that MRR tends to 1 representing almost absolute accuracy even for short queries containing a small number of query patterns. Since a pattern may occur in more than a single song with different prominence like the pattern 2 and 3 given in Table 2 the accuracy of ranking order reflected in MRR improves by adding additional query patterns belonging to the song aimed at. For example the prominence of the pattern 2  $D\#3D\#3C3C3D3D3C3C3C3C3$  is comparatively less in the song aimed at (Aadikondar) and hence reduces the reciprocal ranking and same is the case with pattern 3; however as the query contained three patterns the song aimed at could achieve highest matching score and there by placed in first position in the ranked list for the query of length 3. Figure 3 depicts the graphical representation of MRR versus length of the query in terms of number of query patterns.

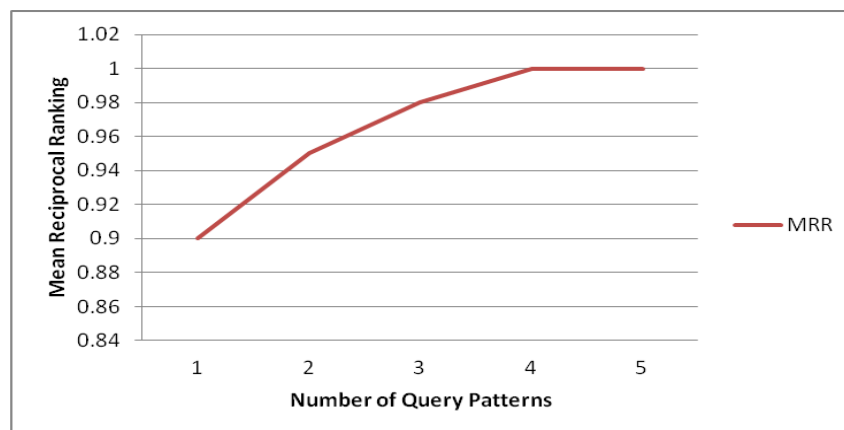


Figure 3: Precision of Music Information Retrieval for different query lengths.



## 5 Conclusion

A framework for content based music information retrieval system that retrieves a ranked list of musical objects from a database in response to query by example is developed. The framework is implemented in terms of five modules for representing the main melody as a note sequence, extracting sequential patterns from the main melody, preparing pattern base of inverted list of musical objects that cover each pattern, query processing and identification of matching patterns using DTW algorithm and finally to rank the music objects after estimating their matching scores to the query expressed as one or more patterns. The established techniques for content based document retrieval were appropriately adopted to build the framework which is tested on real dataset, Raga Surabhi. The framework is found to be very effective as it resulted in very high Mean Reciprocal Ranking (almost one) even for short queries consisting of two to three query patterns.

## ACKNOWLEDGEMENTS

This work was supported by the Council of Scientific & Industrial Research (CSIR) and Andhra University, Visakhapatnam, Andhra Pradesh, India.

## REFERENCES

- [1] Rifki Afina Putri, Dessi Puji Lestari, "Music Information Retrieval using Query-by-Humming based on the Dynamic Time Warping" The 5th International Conference on Electrical Engineering and Informatics 2015 August 10-11, 2015, Bali, Indonesia.
- [2] M. Müller, "Information Retrieval for Music and Motion", New York: Springer, 2007.
- [3] Z. W. Ras and A. Wierzchowska, "Advances in Music Information Retrieval", New York: Springer, 2010.
- [4] Youxi Wu • Shuai Fu • He Jiang • Xindong Wu "Strict approximate pattern matching with general gaps" Springer Science+Business Media New York. 2015
- [5] Anssi Klapuri "Pattern induction and matching in music signals" Exploring Music Contents, Springer 2011
- [6] J. Paulus, M. Muller, and A. Klapuri, "Audio-based music structure analysis," in Proc. of the Int. Society for Music Information Retrieval Conference, Utrecht, Netherlands, 2010.
- [7] J. Serra, E. Gomez, P. Herrera, , and X. Serra, "Chroma binary similarity and local alignment applied to cover song identi\_cation," IEEE Trans. on Audio, Speech, andLanguage Processing, vol. 16, pp. 1138{1152, 2007.
- [8] R. Typke, "Music retrieval based on melodic similarity," Ph.D. dissertaton, Universiteit Utrecht, 2007.
- [9] Alexios Kotsifakos, Isak Karlsson, Panagiotis Papapetrou, Vassilis Athitsos, Dimitrios Gunopulos, "Embedding-based subsequence matching with gaps-range-tolerances: a Query-By-Humming application", The VLDB Journal August 2015, Volume 24, Issue 4, pp 519–536

- [10] D. Vikram, Dr. M. Shashi, "Content Based Indexing of Music Objects using Approximate Sequential Patterns" International Journal of Data Mining & Knowledge Management Process (IJDMP) Vol.5, No.2, March 2015.
- [11] D. Vikram, Dr. M. Shashi, "Content Based Music Information Retrieval: Concepts And Techniques" Journal of Multidisciplinary Engineering Science and Technology (JMEST) ISSN: 3159-0040 Vol. 2 Issue 8, August – 2015
- [12] Hung-Che Shen, Chungnan Lee "Whistle for music: using melody transcription and approximate string matching for content-based query over a MIDI database" Multimed Tools Appl (2007) 35:259–283.
- [13] Raga Surabhi is a collection of audio files containing raga snippets and songs for the process of understanding and learning Carnatic music. <http://www.ragasurabhi.com/>
- [14] J. Stephen Downie, Andreas F. Ehmann, Mert Bay, M. Cameron Jones "The Music Information Retrieval Evaluation eXchange: Some Observations and Insights", Advances in Music Information Retrieval, Springer, 2010.